

In operation, when the ARtryGen block 613' detects a potential deadlocking transaction to a particular slave device, it checks to see if the DLAC bit for that slave device is asserted. If the DLAC bit for that slave is not asserted, then no ARtry signal is generated. If the DLAC bit for that slave device is asserted, then an ARtry signal is generated.

The inputs and outputs of ARtryGen block 613' are illustrated in greater detail in FIG. 26 for the case M=4, S=5 and Q=2.

It will be apparent to those of ordinary skill in the art that the present invention may be embodied in other specific forms without departing from the spirit or essential character thereof. The presently disclosed embodiments are therefore considered in all respects to be illustrative and not restrictive. The scope of the invention is indicated by the appended claims rather than the foregoing description, and all changes which come within the meaning and range of equivalents thereof are intended to be embraced therein.

cm I claim:

1. In a computer system having a system bus and having arbitration circuitry, multiple master devices including a system microprocessor, and multiple slave devices, all coupled to the system bus, a method of reordering system bus transactions, comprising the steps of:

receiving and queuing within a particular slave device a plurality of transactions;

within said arbitration circuitry, arbitrating between pending transactions based on arbitration policies including an arbitration policy that responses are received by respective master devices in the same order as requests were issued by the respective master devices; and at least some of the time, said arbitration circuitry, without signalling said microprocessor, signalling said particular slave device such that the system bus is granted for a later queued transaction within said particular slave device prior to being granted for an earlier queued transaction.

2. The method of claim 1, comprising the further step of maintaining for each master device a master queue in which respective queue entries identify respective target slave devices, and maintaining for each slave device a slave queue in which respective queue entries identify respective originating master devices.

3. The method of claim 2, wherein the step of arbitrating further comprises identifying a winning master device based at least in part on a priority ordering of said master devices, and determining for at least said winning master device a matching queue entry within a slave queue identified by a frontmost queue entry within the master queue of the winning master device, the matching queue entry identifying the winning master device.

4. The method of claim 3, wherein the step of signalling said particular slave device comprises signalling to the particular slave device the matching queue entry identifying the winning master device.

5. The method of claim 4, comprising the further step of the slave devices identifying to the arbitration circuitry pairs of transactions involving the same address block.

6. The method of claim 5, wherein the arbitration circuitry, in identifying the winning master device, ensures that for each pair of transactions identified by the slave devices, a corresponding earlier queued transaction is executed prior to a corresponding later queued transaction.

7. A computer system comprising:

a system bus;

multiple master devices, including a system microprocessor, each coupled to the system bus;

multiple slave devices each coupled to the system bus and each comprising a transaction queue for queuing multiple transactions; and

arbitration circuitry coupled to the system bus and separately coupled to the multiple slave devices for, without signalling said microprocessor, signalling a particular slave device such that within said particular slave device a later queued transaction is executed prior to an earlier queued transaction.

8. The apparatus of claim 7, wherein said arbitration circuitry comprises:

multiple master queues, each corresponding to one of said master devices, in which respective queue entries identify respective target slave devices;

multiple slave queues, each corresponding to one of said slave devices, in which respective queue entries identify respective originating master devices;

means for determining a winning master device based at least in part on a priority ordering of said master devices; and

means for determining for at least the winning master device a matching queue entry within a slave device identified by a frontmost queue entry within the master queue of the winning master device, the matching queue entry identifying the winning master device.

9. An arbiter comprising:

an address arbitration circuit for receiving bus request signals from multiple master devices and in response thereto generating address bus grant signals for the master devices;

a queuing structure including multiple master queues, each corresponding to one of the master devices, and multiple slave queues, each one corresponding to one of multiple slave devices each having a transaction queue, the queuing structure receiving the bus grant signals and receiving respective slave acknowledge signals from respective slave devices, wherein each time an address bus grant is issued a record is entered in the queuing structure, the record comprising a first entry in a master queue identified by the address bus grant signals, the first entry identifying a target slave device in accordance with the slave acknowledge signals, and a second entry in a slave queue identified by the slave acknowledge signals, the second entry identifying an originating master device in accordance with the address bus grant signals;

a matching circuit responsive to queue entries from the queuing structure for producing match bits identifying selected records the first entry of which is at the head of a master queue; and

a data arbitration circuit responsive to the match bits and to queue entries from the queuing structure for generating data bus grant signals for the master devices and for generating for each slave device a multibit signal which when active identifies a transaction within the transaction queue of the slave device.

10. The apparatus of claim 9, wherein said selected records include all records within the queuing structure the first entry of which is at the head of a master queue.

11. The apparatus of claim 10, wherein the match bits partially identify said selected records, entries at the head of the master queues being used in combination with the match bits to uniquely identify the selected records.

(14)

27

12. The apparatus of claim 11, wherein the matching circuit is responsive to read-ready signals from the slave devices for producing read-ready bits in one-to-one correspondence with the match bits.

13. The apparatus of claim 12, wherein the matching circuit produces a match bit and a read-ready bit for each queue location of the slave device transaction queues.

14. The apparatus of claim 12, wherein the data arbitration circuit produces a signal bit for each queue location of the slave device transaction queues.

15. The apparatus of claim 12, wherein the data arbitration circuit comprises a bit filter and is responsive to address

28

coincidence signals from the slave devices for filtering the match bits prior to selecting a winning master device.

16. The apparatus of claim 15, wherein the address coincidence signals identify pairs of transactions involving the same block of addresses.

17. The apparatus of claim 16, wherein the data arbitration circuit ensures that for each pair of transactions identified by the slave devices, a corresponding earlier queued transaction is executed prior to a corresponding later queued transaction.

\* \* \* \* \*

Ack  
C2

15